Supplementary Information for "Reverse engineering the control law for schooling in zebrafish using virtual reality"

Liang Li[1,2,3*], Máté Nagy[1,2,3,4,5,6*], Guy Amichay[1,2,3,7], Wei Wang[8], Oliver Deussen[2,9], Daniela Rus[8], and Iain D. Couzin[1,2,3*]

[1] *Department of Collective Behaviour, Max Planck Institute of Animal Behavior, 78464 Konstanz, Germany*

[2] *Centre for the Advanced Study of Collective Behaviour, University of Konstanz, 78464 Konstanz, Germany*

[3] *Department of Biology, University of Konstanz, 78464 Konstanz, Germany*

[4] *MTA-ELTE "Lendület" Collective Behaviour Research Group, Hungarian Academy of Sciences, 1117 Budapest, Hungary*

[5] *Department of Biological Physics, Eötvös Loránd University, 1117 Budapest, Hungary*

[6] *MTA-ELTE Statistical and Biological Physics Research Group, Eötvös Loránd Research Network, 1117 Budapest, Hungary*

[7] *Department of Engineering Sciences and Applied Mathematics, Northwestern University, Evanston, IL, USA*

[8] *Computer Science and Artificial Intelligence Lab (CSAIL), Massachusetts Institute of Technology, Cambridge, MA, USA*

[9] *Department of Computer and Information Science, University of Konstanz, 78464, Konstanz, Germany*

Supplementary Note 1

Experiments with 2 real fish in the same arena and in the Matrix.

We also conducted 'traditional' experiments (i.e., those in which real fish swim together) in the same virtual reality arena with the same background projection, but without any virtual fish. This allowed us to analyze data from two real fish swimming in the 3D environment. We calculated and compared the lateral and forward swimming speeds of the following fish to those obtained in the virtual reality system. We next analyzed the lateral and forward speeds of the follower whenever there were leader-follower patterns (see filter and parameters below). Lateral and forward speeds were defined in relation to a coordinate system centered on the leader, with the positive direction pointing towards the leader's head (estimated as swimming direction, Fig. 2a). We averaged the speed over 1 second to obtain the average swimming speed. We divided the speed data into a heatmap with 30 grid cells in the left-right distance and 15 grid cells in the front-back distance. We colored each bin based on the average swimming speed within it.

In our analyses of the leader's swimming properties, we applied an algorithm to identify bursts of high swimming speeds and extract burst-glide swimming periods. Maximum swimming speeds over the extracted burst-glide periods are defined as v_{max}, and the time to reach the maximum swimming speeds is defined as $t_{v_{max}}$ (Supplementary Fig. 7i-k).

Experiments with 1 virtual fish swimming at different average speeds.

In these experiments, we controlled the virtual fish to swim back-and-forth in the arena at a depth of 0.03 m and along a straight line of length of 0.24 m (Fig. 2b). The average speed of the virtual fish ranges from 0.04 to 0.08 m/s with an interval of 0.01 m/s under burst-and-glide motion extracted from a random real fish's swimming pattern (average swimming speed is 0.04 m/s and the length of the segment is 8 seconds, Fig. 2c). Faster speeds of the virtual fish are achieved by multiplying the base speed by a factor of (1.25, 1.5, 1.75, and 2). The stable distance lags between the leader, and follower are determined based on the relative position of the follower at a coordinate determined based on the leader (Fig. 2a). The lateral and forward speeds of the follower are calculated as we did for the two real fish case (Fig. 2e, f). Average lateral/forward speed is calculated based on the heatmap along the front-back/left-right direction. The critical turning point of the speed $r_x(r_y)$ is determined based on peak detection. Bootstraps are applied to obtain the statistics of the lateral/forward swimming speeds and the turning points. The parameter K_d is determined by the ratio of the line, which describes the forward swimming speed of the real fish at the critical turning point r_x as a function of the average swimming speed of the leader (see detailed mathematical derivation below). To determine K_p, we simulate (without noise) the follower following a leader swimming at the corresponding average swimming speed. We simulate K_p ranges from 1.2 to 8 with an interval of 0.1. The K_p is optimized to minimize the accumulated differences between the leader-follower distance lags in simulations and experiments.

1 virtual fish swimming with different swimming patterns.

In these experiments, we control the virtual fish swimming with different patterns— burst-and-glide and constant speed (Fig. 3a and Supplementary Fig. 10) —but with the same average values (0.04, 0.05, 0.06, 0.07, and 0.08 m/s). The other experimental setting and analyses are the same as described above.

1 virtual fish swimming with different refresh frequencies.

In this experiment, we varied the refresh frequency of the virtual fish. The swimming speed was set at 0.04 m/s and the fish was programmed to swim with a burst-and-glide pattern. We set the refresh frequency at 100, 50, 20, 10, 5, or 1 Hz. The refresh frequency determines how frequently we update the virtual fish, which does not affect the average swimming speed. We analyzed the distance lags and lateral/forward swimming speeds in the same way as described above.

1 virtual fish with different visibilities

We varied the visibility of the virtual fish by showing it for short periods of time (0.2 s). As a control, we set the virtual fish to be invisible for short periods of time (0.2 s or 0.4 s) and move at the same or double the original speed only when in this invisible state. We set the appearance time to 0.2 seconds to avoid flashing virtual fish. The rest of the experiments and data analyses were the same as described above.

2 real fish in the matrix with BioPD

In these experiments, we controlled the virtual fish in one arena (arena A) based on the behavior of the real fish in the other arena (arena B). While in arena B, we controlled the virtual fish according to the real fish in arena A only when the real fish was swimming behind the virtual fish. Otherwise, the virtual fish was controlled by the BioPD algorithm. We analyzed data from both arenas, as in one arena the real fish followed the virtual fish, and in the other arena, the virtual fish followed the real fish.

2 virtual fish swimming side by side at different average speeds

In these experiments, we controlled two virtual fish swimming side by side and back and forth within the bowl. We varied the left-right distances and average swimming speeds to establish how the real fish controlled its following behavior. The left-right distance was set to 0.01 to 0.12 m in intervals of 0.005 m. The average swimming speed varied from 0.04 to 0.08 m/s in intervals of 0.01 m/s. We analyzed the distance lag of the following real fish in the local coordinate system of the two virtual fish, where the center of the two virtual fish is the origin and the positive y axis is along the virtual fish's heading direction (Fig. 4j). We averaged the distance lag along the left-right distance and plotted it in a heatmap. Similarly, we also averaged the heatmap of the following behavior along the front-back distance to show the decision-making in the x-axis (Supplementary Fig. 15).

Supplementary Note 2

Parameter estimation for the model

Our model modifies the traditional PD controller by adding a nonlinear relationship (described by a first-order Gaussian derivative function) between speed and relative position. This function accounts for the phenomenon that when the distance is larger than a threshold distance r_x (r_y), the speed reduces and the follower will reduce its speed to follow the leader. The threshold distance r_x (r_y) corresponds to the peak of the first-order Gaussian derivative function. In other words, the follower should be within the distance threshold to catch up with the leader, and if it is, the performance is similar to the traditional controller. We first theoretically analyze the final stable distance by following derivatives: assume a follower is following a leader on the y-axis; the relative position in the y-axis is Δy. The controller of the follower with a traditional PD controller is:

$$v_{RF} = K_p \cdot \Delta y + K_d \cdot \Delta \dot{y} \tag{1}$$

Then the variation of the distance lag is:

$$\Delta \dot{y} = \dot{y}_{VF} - \dot{y}_{RF} = v_{VF} - v_{RF} \tag{2}$$

Substituting Eq. 1 into Eq. 2 yields:

$$\Delta \dot{y} = v_{VF} - v_{RF} = v_{VF} - K_p \cdot \Delta y - K_d \cdot \Delta \dot{y} \tag{3}$$

Solving this equation, we get:

$$\Delta y = \frac{v_{VF}}{K_p} + \frac{e^{\frac{K_p(C_1 - t)}{K_d + 1}}}{K_p} \tag{4}$$

Where C_1 is a constant value depending on the initial condition. The final stable distance ($t \to \infty$) will be:

$$\Delta y_{t \to \infty} = \frac{v_{VF}}{K_p} \tag{5}$$

In another way, since $v_{RF} = K_p \cdot \Delta y + K_d \cdot \Delta \dot{y}$ and $\dot{y} = \dot{y}_{RF} - \dot{y}_{VF} = v_{RF} - v_{VF}$, we can rewrite the virtual fish speed as:

$$v_{RF} = K_p \Delta y + K_d (v_{RF} - v_{VF}) \tag{6}$$

Therefore, we get the description of real fish swim speed as:

$$v_{RF} = \frac{K_d}{K_d - 1} v_{VF} + \frac{K_p}{1 - K_d} \Delta y \tag{7}$$

Therefore, we obtained the relationship (first order) between the real fish's swimming speed and the virtual fish's swimming speed. Using bootstrapping, we resampled lateral and forward swimming speeds to get the heatmaps (see Fig. 2e, f in the main text). According to Eq. 7, if Δy is fixed, the relationship between real and virtual fish speeds will be linearly correlated. The ratio is a function of K_d. We then fit the swimming speed of the real fish at the critical distance ($r_x = r_y = 0.07$ m) as a function of the average speed of the virtual fish to obtain the ratio k. Based on the ratio k and Eq. 7, we calculated $K_d = \frac{k}{k-1}$.

The model for 2 virtual fish as leaders

$$
\begin{cases}
x_{e1} & = & x_F - x_{L1} \\
x_{e2} & = & x_F - x_{L2} \\
v_x & = & -\left[(K_p x_{e1} + K_d \dot{x}_{e1}) e^{-\frac{x_{e1}^2}{2r_x^2}} + (K_p x_{e2} + K_d \dot{x}_{e2}) e^{-\frac{x_{e2}^2}{2r_x^2}}\right] \\
y_e & = & y_F - y_L \\
v_y & = & -(K_p y_e + K_d \dot{y}_e) e^{-\frac{y_e^2}{2r_y^2}} \\
r_x & = & a \cdot y_e + b
\end{cases}
\tag{8}
$$

where x_{L1} and x_{L2} are the distance differences between the left and right virtual fish and the real fish in a coordinate system determined based on the two virtual fish (Fig. 4j). y_L is the distance difference in the y-axis in the local coordinate system. Because $y_{L1} = y_{L2}$, we set one distance lag y_L for both individuals. Due to the linear perspective, fish perceive the distance in the x-axis to decrease as the distance in the y-axis increases. Parameters a and b are determined based on the fitted critical point in the bifurcation (Supplementary Fig. 15). We fit the bifurcation according to the following steps:

We extract the peaks of the data on the x-axis and then fold the values according to $x = 0$. We then fit a piecewise function to determine the critical point l_c:

$$
x_l = \begin{cases}
0 & l \leq l_c \\
\alpha |l - l_c|^{\beta} & l > l_c
\end{cases}
\tag{9}
$$

where α, β, l_c are fitting parameters, x_l is the detected peak value in the x-axis.

To fit the parameters in the linear perspective model, we first determine the distance lags (d)

at different average swimming speeds based on the distributions (Supplementary Fig. 12).

We then fit a piecewise function to determine the critical bifurcation points (l_c). We use these points, along with the distance lag, d, to determine the parameters (a, b) in the linear perspective model.

$$l_c = 0.28d + 0.05 \tag{10}$$

Because the distance lag, d, is linearly correlated to the average swimming speeds of the leader, v, the relationship between l_c and v is also linear and can be described as:

$$l_c = 0.27v + 0.04 \tag{11}$$

We maintained the value of K_d from the 1 virtual fish case and fit K_p based on the observed distance lag in the experiments, resulting in a value of $K_p = 1.9$.

Supplementary Note 3

Apply BioPD in multiple robotic platforms (terrestrial, airborne, and watercraft):

In these experiments, we used BioPD and an optimal controller to control three different types of robots: the Crazyflie drone [1], the SunFounders Robot PiCar-X, and a robot boat from MIT [2]. We created a virtual leader that moved in a sinusoidal curve for the robots to follow, and we implemented BioPD and a model predictive controller (MPC) based on our previous research. We implemented both BioPD and model predictive controller (MPC) based on our previous study [2].

In order to implement the BioPD controller for the drone, we provided velocity in the x- and y-axes. We used the same values for the proportional (K_p =2.3) and derivative (K_d =0.58) controls. Since the size of the robot is larger than the size of the real fish (1 cm), we scaled the average swimming speed of the leader by a factor of 2 to increase the safety distance. The turning point of the speed control r_x and r_y was also scaled by a factor of 2. The position of the robot was tracked using a Qualisys motion capture system. With MPC, we designed the state as $\boldsymbol{q} = [\boldsymbol{x}, \boldsymbol{y}]^{\boldsymbol{T}}$, and considered a simple kinematic model:

$\dot{\boldsymbol{q}} = \boldsymbol{u}$, where $u = [v_x, v_y]$. We apply the controller every sampling instant by solving the following open-loop optimal controller with a finite horizon: $\underset{u(\tau)}{min} = J(q(\tau), u(\tau))$ subject to $\dot{\mathbf{q}}(\tau) = \mathbf{u}(\tau), \mathbf{q}(\mathbf{0}) = \mathbf{q_0}$, where $\mathbf{u}_\tau \in \mathbf{U}, \forall \tau \in [\mathbf{t}, \mathbf{t} + \mathbf{T}]$. T is the horizon in the controller, and $J(q(\tau), u(\tau))$ denotes the objective function and is described as $J(q(\tau), u(\tau)) = \int_t^{t+T} F(\mathbf{u}(\tau)) + \mathbf{E}(\mathbf{q}(\mathbf{t} + \mathbf{T}))$, where F is the cost function regarding the desired performance objective, and E is the terminal cost (See [2] for details).

Since the robot car cannot move sideways, we cannot directly apply BioPD to control v_x and v_y. Instead, we convert these two speeds into speed $v = \sqrt{v_x^2 + v_y^2}$ and angular speed $\omega = \dot{\theta}$, where $\theta = \arctan(v_y/v_x)$. The state in the MPC here is $\mathbf{q} = [\mathbf{x}, \mathbf{y}, \theta]^{\mathbf{T}}$. Control energy is estimated by $\int_0^t \mathbf{u}(\tau)^{\mathbf{T}}\mathbf{u}(\tau)$.

Supplementary Note 4

The filter for the two real fish swimming in the same arena and the Matrix system. We apply the following filters for the data analyses for both cases.

- Two fish swim within 0.005 to 0.2 meters.
- The distance difference in the z-axis is within 0.03 meters.
- Two fish keep these relationships at least for 1 second.
- Fig. S3 shows one example of two fish swimming in leader-follower cases.

The filter for the data with one virtual fish swimming in the arena.

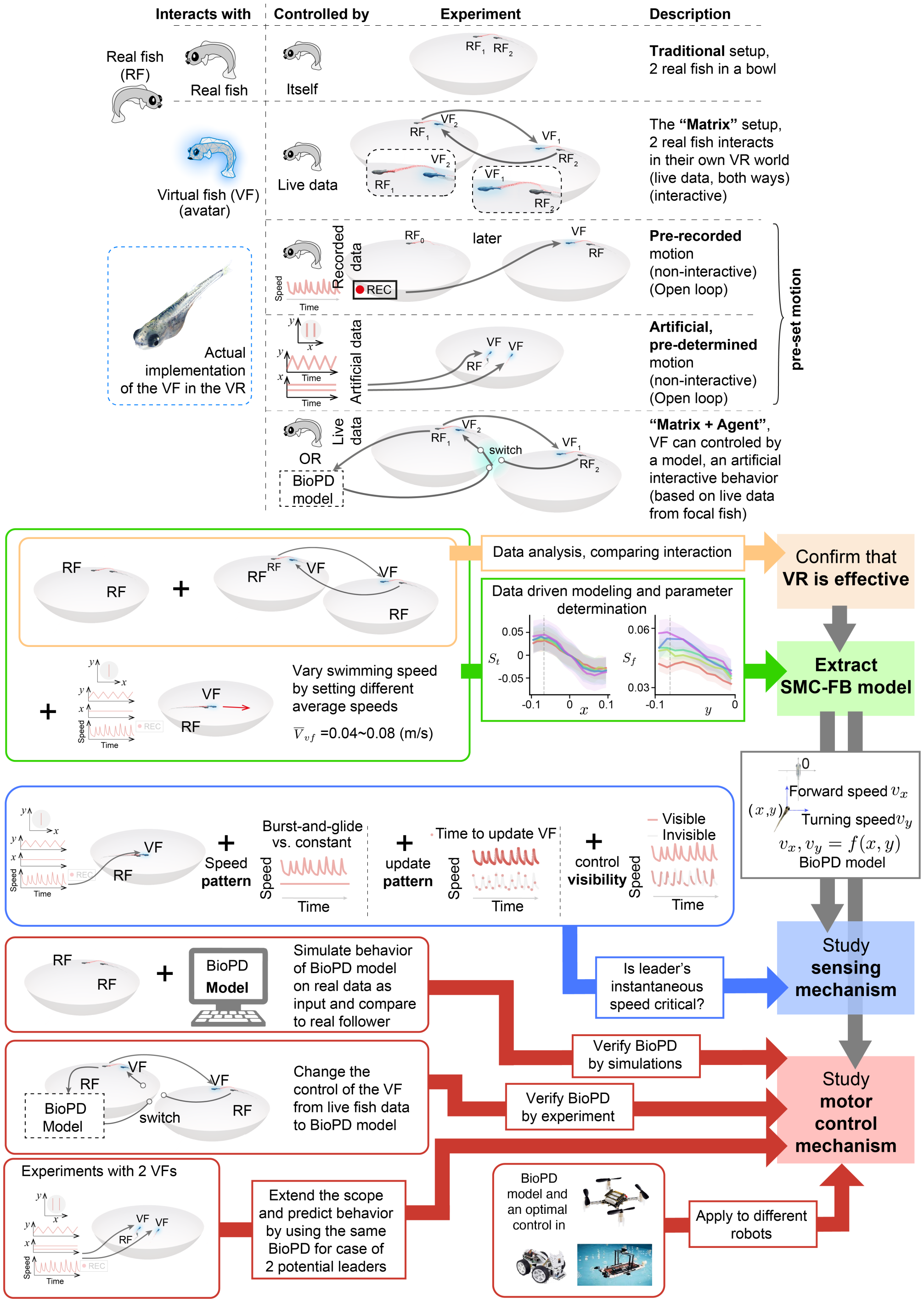
In order to analyze the distance lag, lateral speed, and forward speed, we applied filters to the data. The filters used for the distance lag analysis and the speed along the x- and y-axes are slightly different because we are primarily interested in the relative stable following distance lag. We added a swimming direction difference filter to the distance lag analysis and shortened the continuous frames to 30 frames (0.3 seconds), which is a typical burst-gliding swimming period for our fish (Supplementary Fig. 7).

We apply the following filters for the data analyses for both cases.

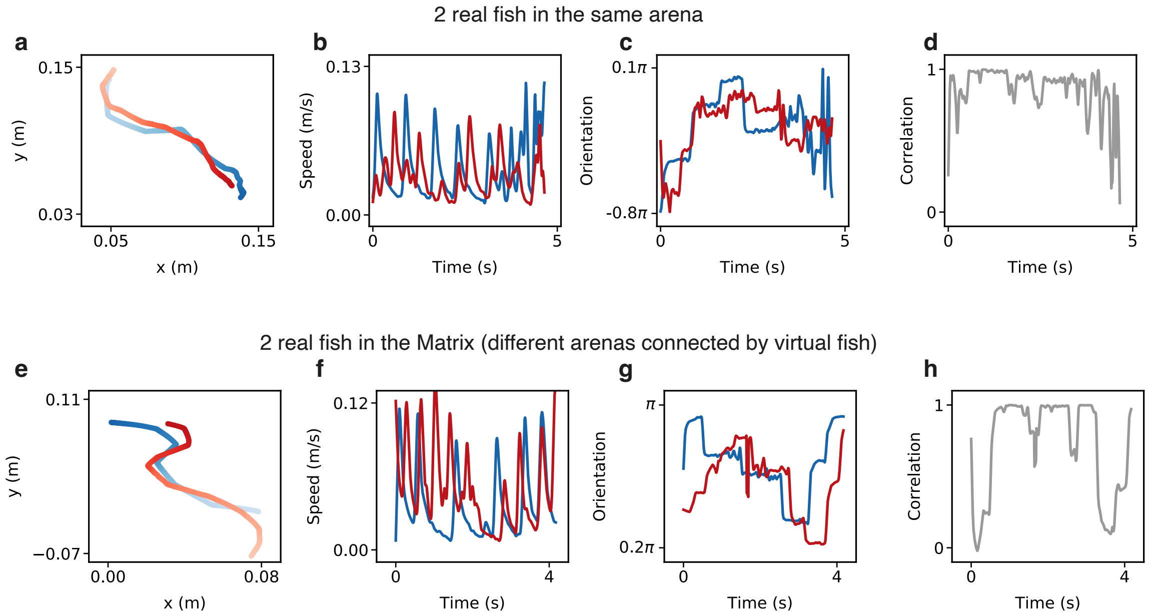
- Two fish swim within 0.2 meters in x- and y- axes.
- Real fish is behind the virtual fish.

- Swimming direction differences between leader and follower are within 30 degrees.

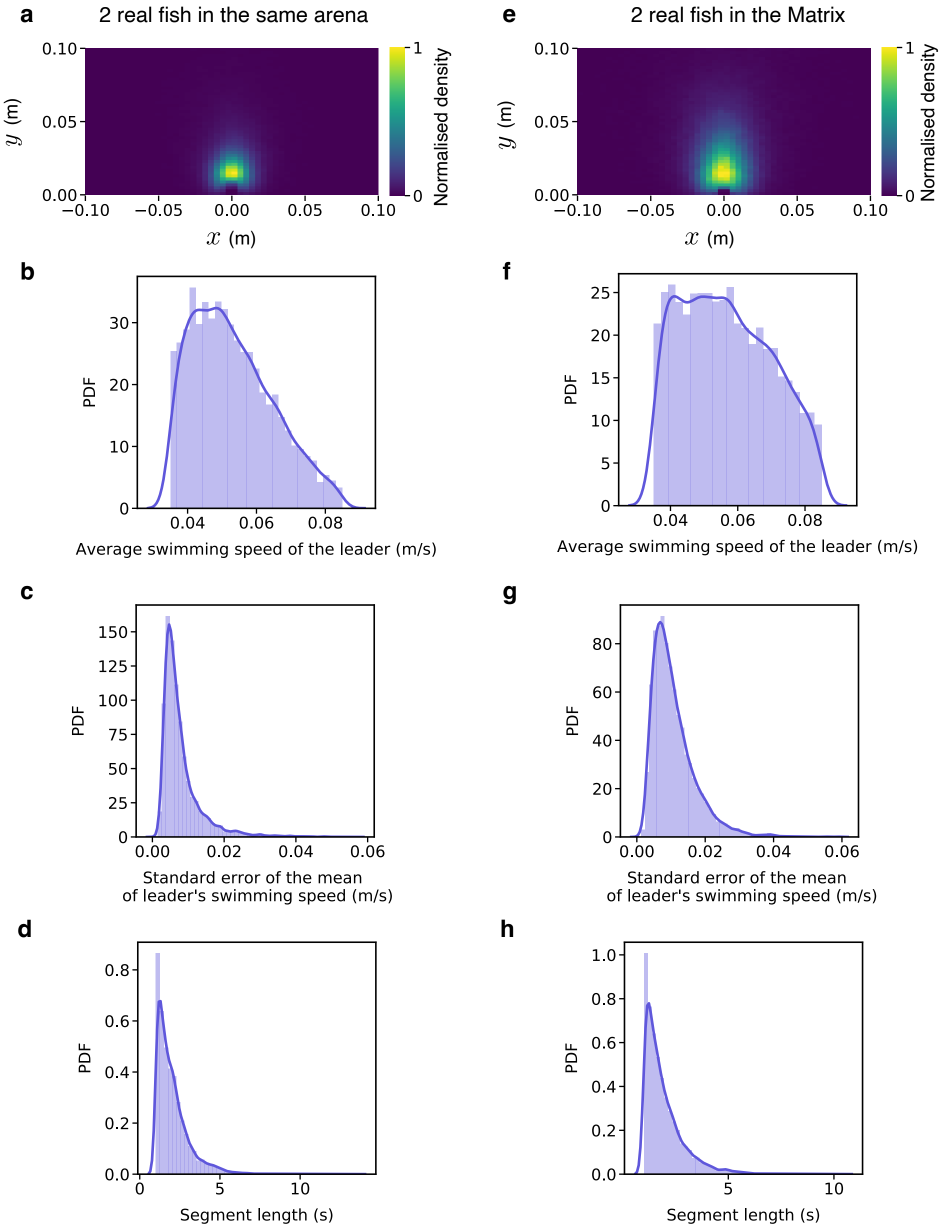
- Virtual fish is in the center of the bowl with a radius of 0.1 m. Two fish keep these relationships for at least 0.3 seconds.


Interacts with
Controlled by
Experiment
Description
Real fish (RF)
Real fish
Itself
RF1 RF2
Traditional setup, 2 real fish in a bowl
Virtual fish (VF) (avatar)
Live data
VF2 RF1 VF1 RF2
The "Matrix" setup, 2 real fish interacts in their own VR world (live data, both ways) (interactive)
Actual implementation of the VF in the VR
Recorded data
Speed
Time
REC
RF0
later
VF
RF
Pre-recorded motion (non-interactive) (Open loop)
pre-set motion
Artificial data
y
x
Time
VF VF RF
Artificial, pre-determined motion (non-interactive) (Open loop)
Live data
OR
BioPD model
switch
"Matrix + Agent", VF can controled by a model, an artificial interactive behavior (based on live data from focal fish)
RF RF + RF RF VF VF RF
Data analysis, comparing interaction
Confirm that VR is effective
Data driven modeling and parameter determination
S_t
S_f
0.05
0
-0.05
0.06
0.03
-0.1
0
0.1
x
y
Extract SMC-FB model
+ VF RF
Vary swimming speed by setting different average speeds
$\bar{V}_{vf}$ =0.04~0.08 (m/s)
0
Forward speed v_x
(x,y)
Turning speedv_y
$v_x, v_y = f(x, y)$
BioPD model
Burst-and-glide vs. constant
+ Speed pattern
Time to update VF
+ update pattern
Visible
Invisible
+ control visibility
Is leader's instantaneous speed critical?
Study sensing mechanism
RF RF + BioPD Model
Simulate behavior of BioPD model on real data as input and compare to real follower
Verify BioPD by simulations
Change the control of the VF from live fish data to BioPD model
Verify BioPD by experiment
Study motor control mechanism
Experiments with 2 VFs
Extend the scope and predict behavior by using the same BioPD for case of 2 potential leaders
BioPD model and an optimal control in
Apply to different robots

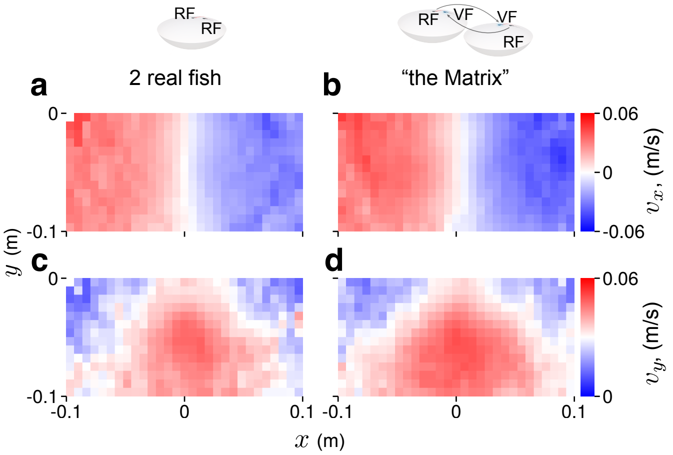
SUPPLEMENTARY FIGURE 1. Outline of this study. We justify and apply a novel virtual reality platform to reverse engineer a simple biological-inspired proportional derivate control, which describes sensory-motor control of zebrafish following behavior. With the VR platform, we further checked the model assumption and verified the control. Finally, we applied the BioPD control to three different types of robots.



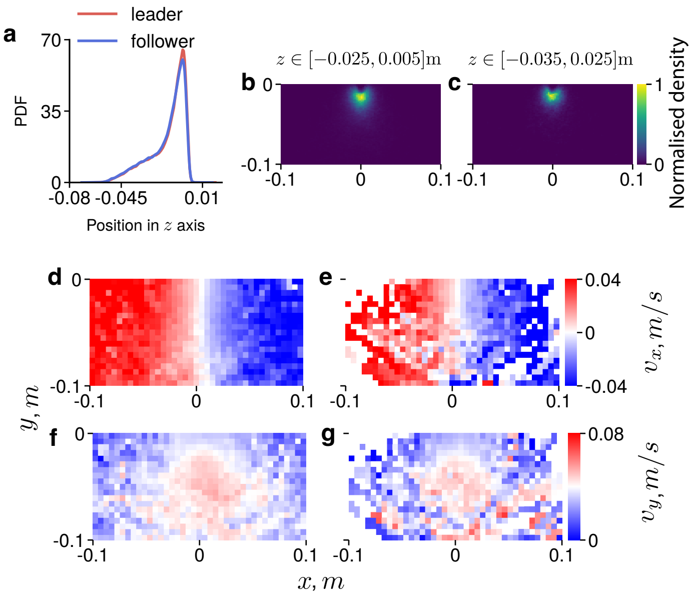
SUPPLEMENTARY FIGURE 2. Leader-follower swimming trajectories extracted from the experiments in the same arena or "the Matrix" system. Trajectories (**a**, **e**) swimming speed (**b**, **f**) orientation (**c**, **g**) and correlation (**d**, **h**) of two fish swimming in the same arena (**a-d**) and "the Matrix" system (**e-h**).



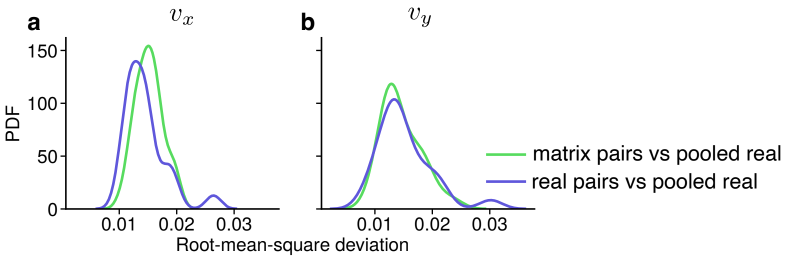
SUPPLEMENTARY FIGURE 3. Probability density distributions (PDF) of swimming properties of two fish swimming in the same arena or “the Matrix”. **a**, **e**, PDF of the leader’s position around the focal individual. **b**, **f**, Comparison of leader’s swimming speed. **c**, **g**, Comparison of the standard error of the leader’s average swimming speed. **d**, **h**, The length of the segments of the pairs.



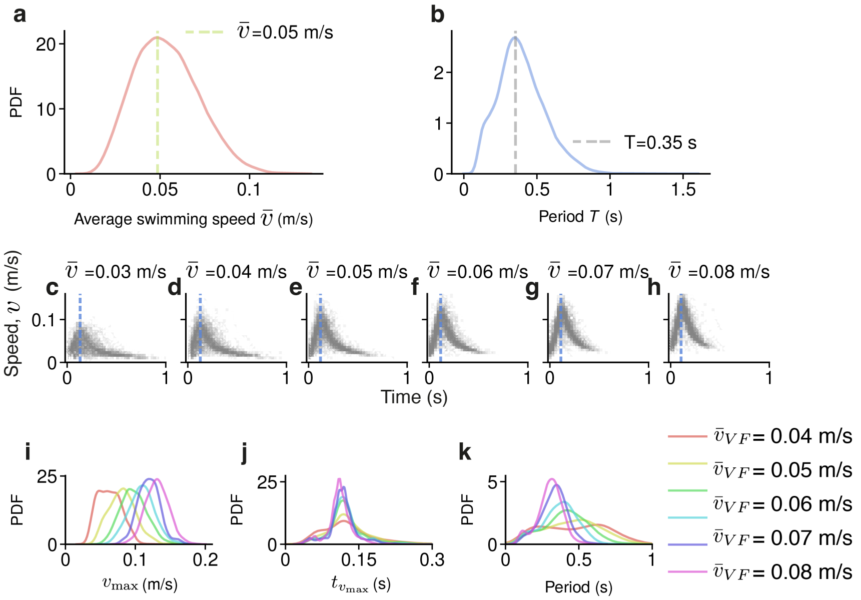
SUPPLEMENTARY FIGURE 4. The comparisons of the lateral, v_x (a, b) and forward, v_y (c, d) speed as a function of the relative positions of the real or virtual leaders. The relative position of the leader is binned according to the focal individual, and we only consider how the focal fish (the follower) reacts to the neighbor who is swimming in front (the leader).



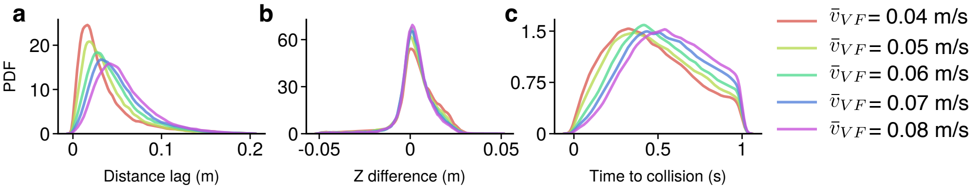
SUPPLEMENTARY FIGURE 5. Social interaction in the same depth shows similar lateral and forward speeds relative to the leader. a, Distributions of two real fish in z axis. **b-c**, Position density of the follower at different ranges of depth. Lateral speed v_x (**d**, **e**), and forward speed v_y (**f**, **g**), as a function of the follower's position relative to the leader at different ranges of depth.



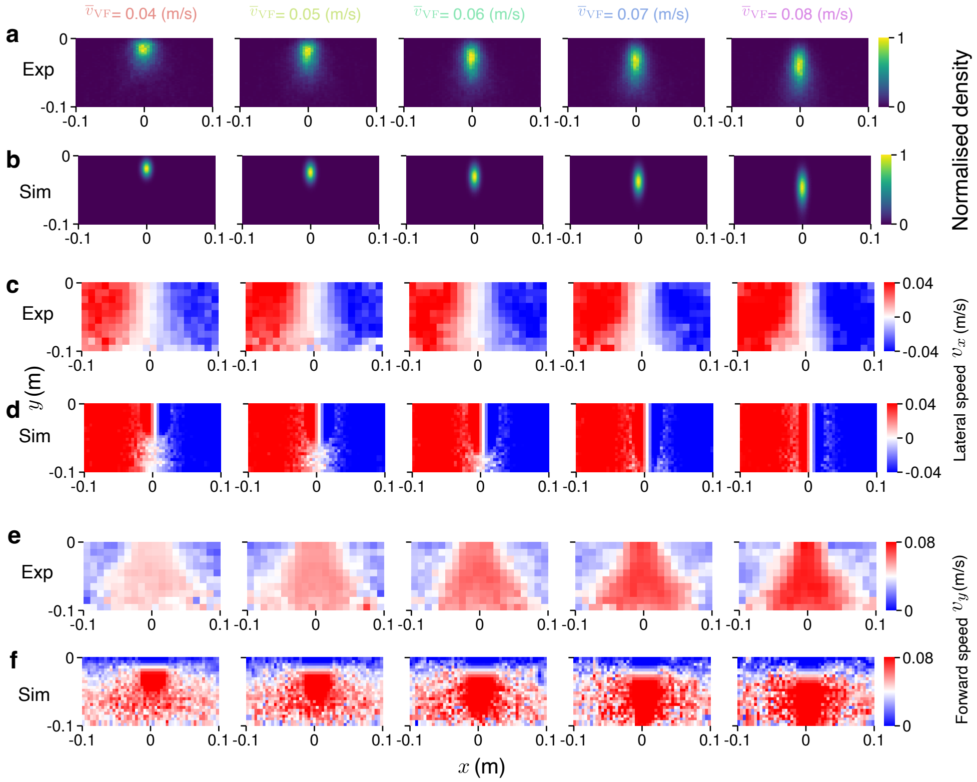
SUPPLEMENTARY FIGURE 6. Root-mean-square-deviation (RMSD) of the lateral (a) and forward (b) speeds between pairs (real and "the Matrix") and pooled real interaction data. We pooled 22 pairs of real interaction data as a reference and compared the remaining 24 pairs of real interaction data and 24 pairs of virtual (or "Matrix") data to the reference. We found that there was no significant difference between the real and virtual data, as determined by Kolmogorov–Smirnov tests (p=0.26 for lateral speed, p=0.9 for forward speed).



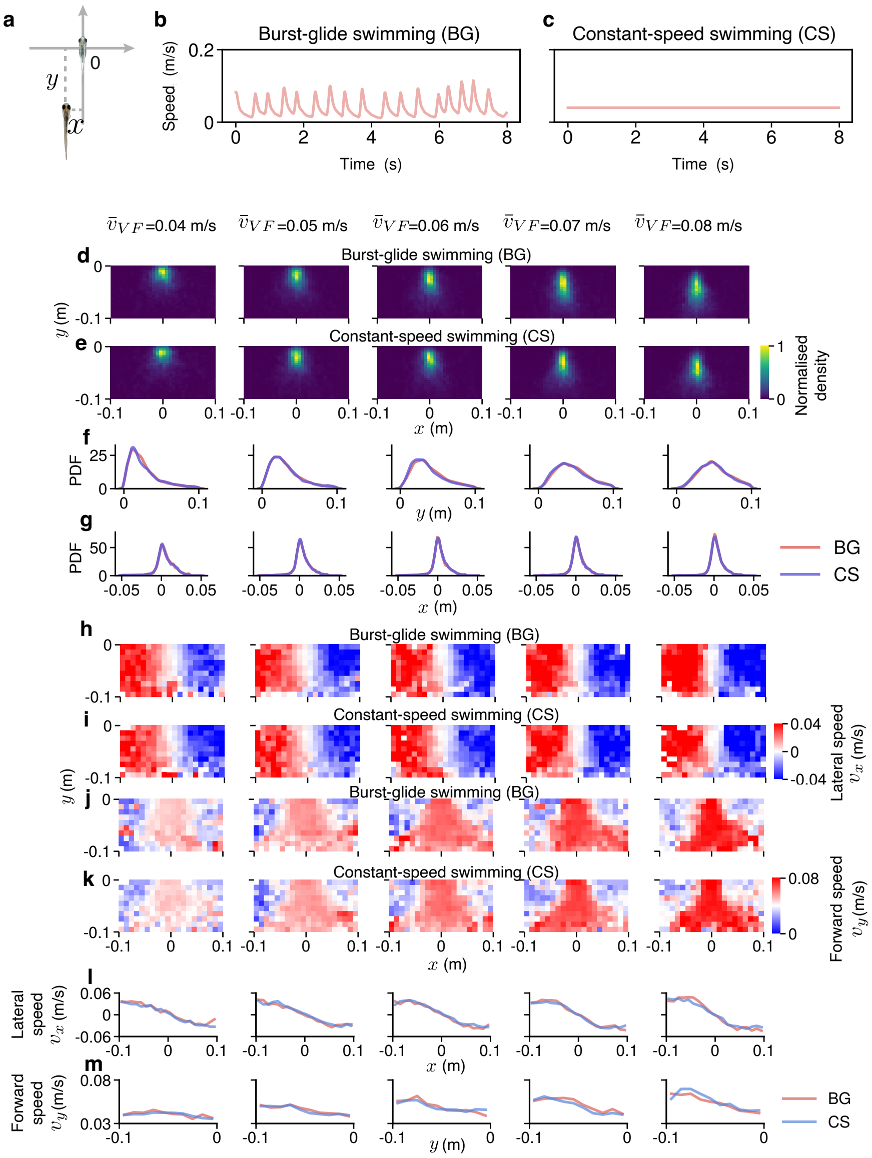
SUPPLEMENTARY FIGURE 7. Swimming properties of the leader. a, Distribution of average swimming speed of the leader over one burst-and-glide period. The typical swimming speed of a leader is around 0.05 m/s. **b**, Distributions of the periods of the burst-and-glide swimming. The typical period is around 0.35 seconds. **c**-**h**, 500 examples of the burst-and-glide swimming pattern with different average swimming speeds. **i**, Distributions of the maximum swimming speed under different average swimming speeds. **j**, Distributions of the time when the leader reaches maximum swimming speed. **k**, Distribution of the periods when the leader swims with different average swimming speeds.



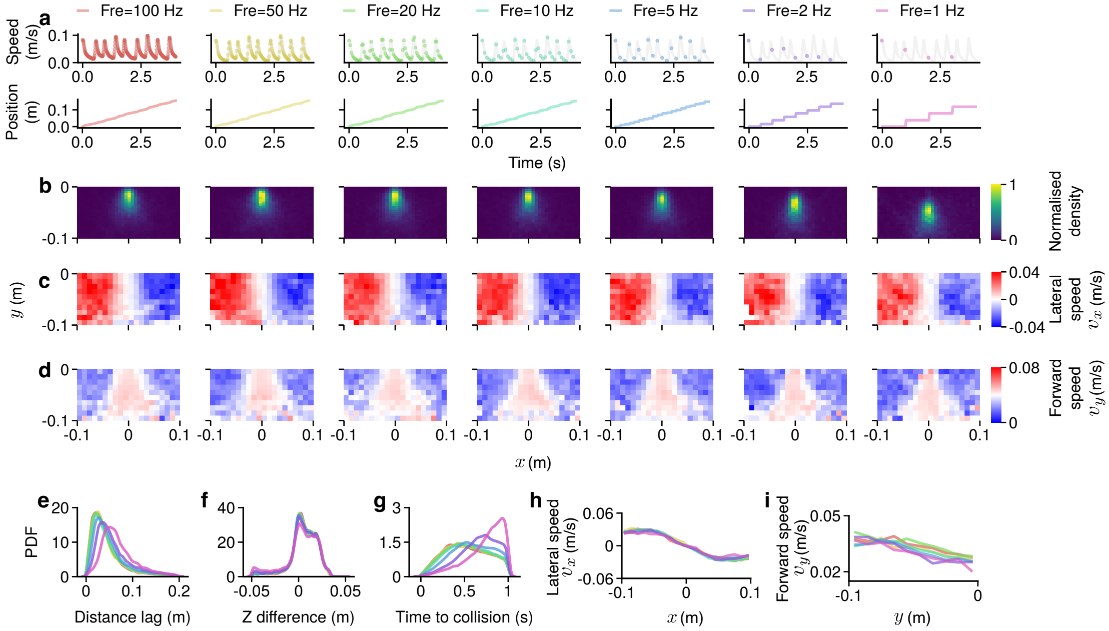
SUPPLEMENTARY FIGURE 8. A comparison of (a) distance lag, (b) difference in depth in the water column (z axis), and (c) "time to collision" with respect to the current position of the leader (if the leader were to suddenly stop)



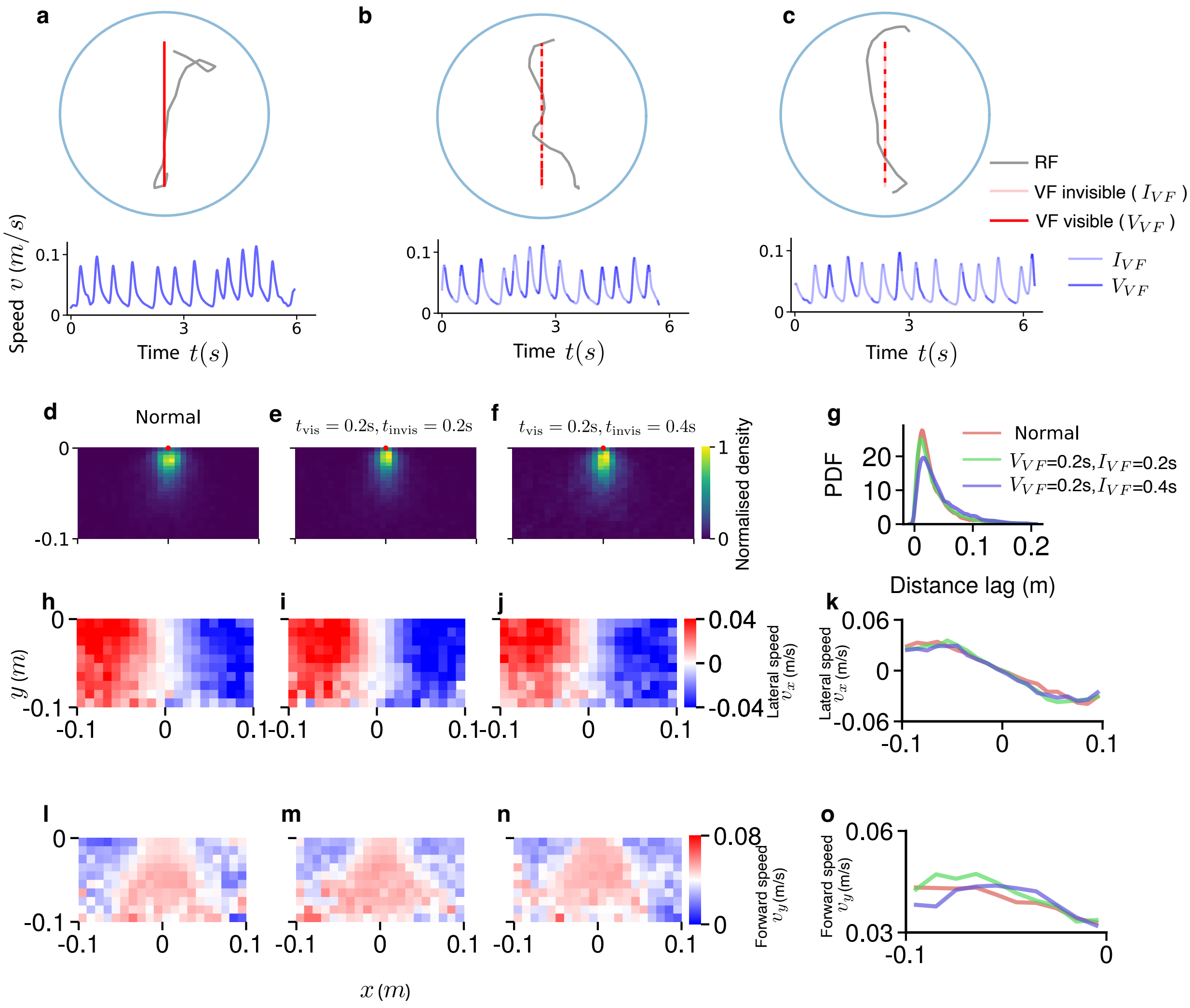
SUPPLEMENTARY FIGURE 9. Relative position, lateral speed, and forward speed of the follower in experiments (Exp, a, c, e) and simulations (Sim, b, d, f). We initialized the follower's position over a range from -0.05 to 0.05 m on the x-axis and -0.05 to 0 m on the y-axis. White noise was added to the follower's speed control, with a standard variance of 0.016 for the x-axis and 0.45 times the average speed of the follower for the y-axis. The maximum swimming speed was limited to 0.1 m/s. The position density, lateral speed, and forward speed of the follower were then analyzed using the same methods as in the experiments.



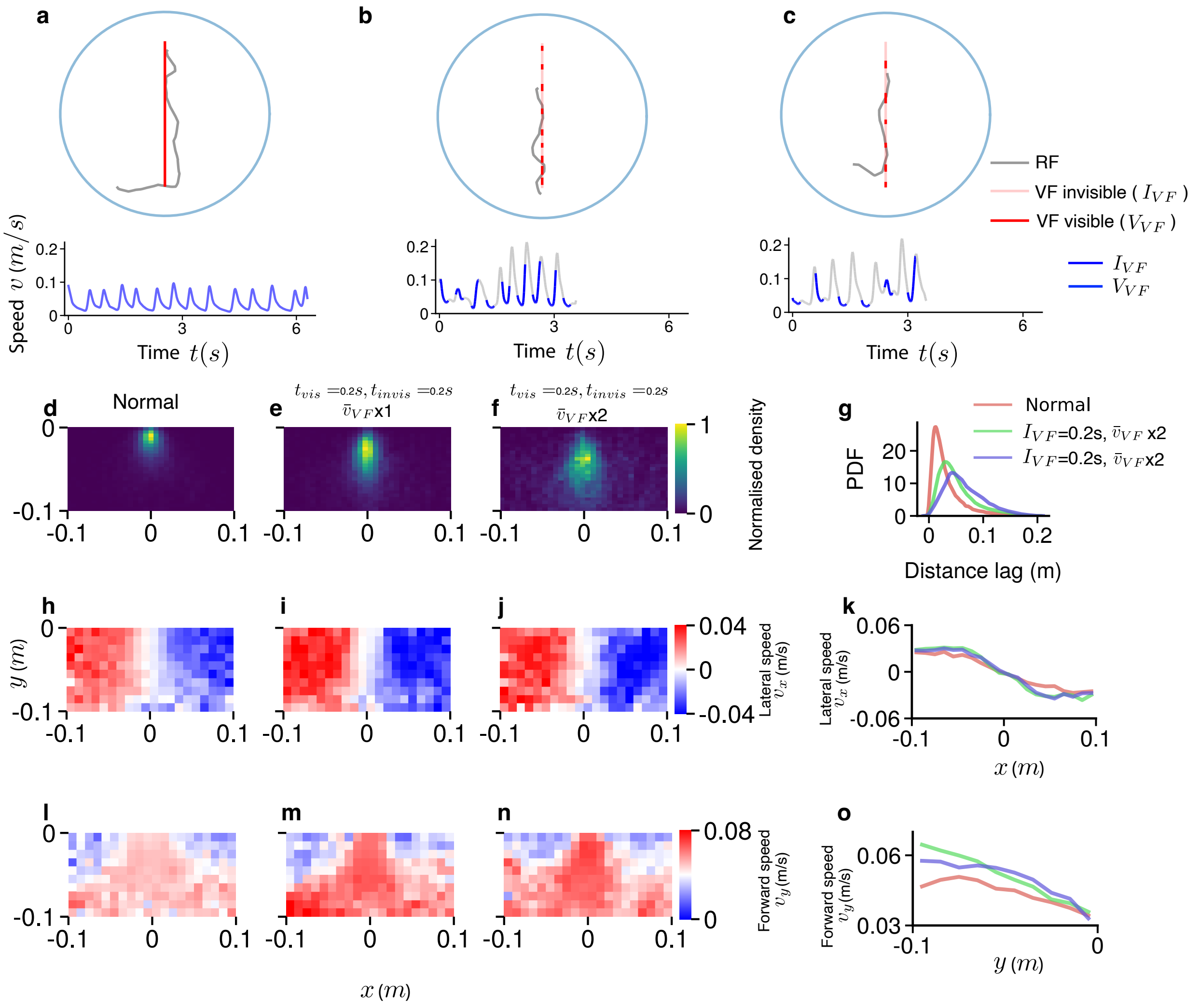
SUPPLEMENTARY FIGURE 10. Behaviors of the real fish following a leader swimming with burst-and-glide patterns and constant speed patterns. a, the coordinate system. **b**, Example of the burst-and-glide (BG) swimming pattern at an average swimming speed of 0.04 m/s. **c** Constant swimming (CS) pattern with an average swimming speed of 0.04 m/s. **d-g**, Positions of the follower relative to the leader swimming with BG (**d**, **f**) or CS (**e**, **g**) patterns. **h-m**, Lateral and forward swimming speed as a function of the neighbor's position with the leader swimming with BG (**h**, **j**, **l**) and CS (**i**, **k**, **m**) patterns.



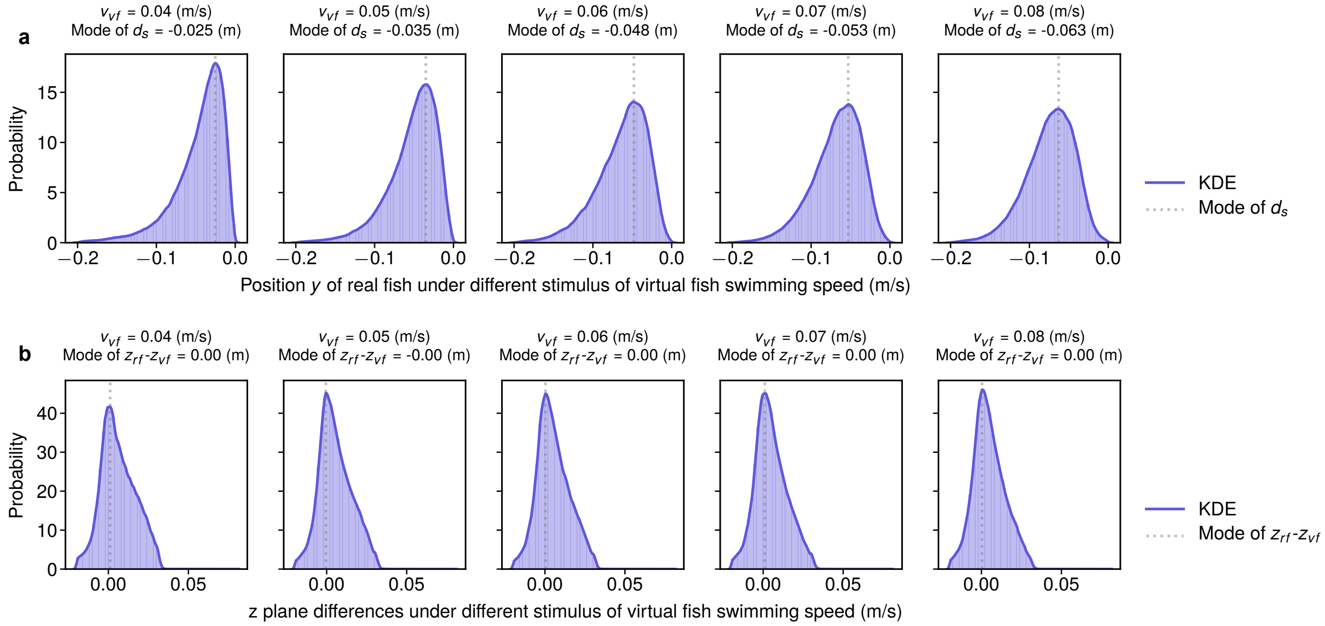
SUPPLEMENTARY FIGURE 11. Behaviors of the real fish following a leader with different refresh frequencies. a, Illustration of the frequency of updating virtual fish swimming as a leader at speed 0.04 m/s and corresponding virtual fish position. **b** Positions of the follower relative to the leader. **c-d**, Lateral (**c**) and forward (**d**) swimming speed of the follower as a function of the leader's position. **e-f**, Distribution of the distance lag (**e**) and difference in z-plane (**f**) between the leader and the follower. **g**, "Time to collision" for the follower. **h**, Lateral and (**i**) forward swimming speed of the follower as a function of the neighbor's position.



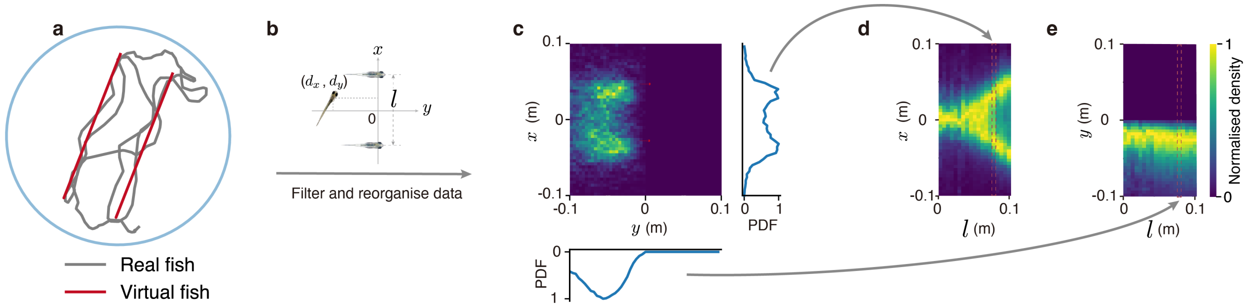
SUPPLEMENTARY FIGURE 12. Behaviors of the real fish following a leader with varying visibility. a-c, Illustration of the visibility control of the virtual fish swimming as a leader at a speed of 0.04 m/s. **d-g**, Positions of the follower relative to the leader. Lateral (**h-k**) and forward (**l-o**) swimming speed of the follower as a function of the leader's position.



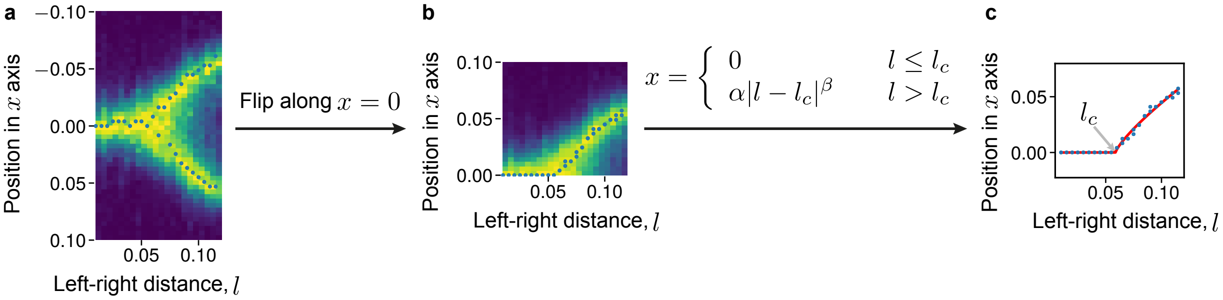
SUPPLEMENTARY FIGURE 13. Behaviors of the real fish following a leader with varying visibility and speeds of the virtual fish. a-c, Illustration of the visibility and speed of the virtual fish swimming as a leader at a speed of 0.04 m/s. **d-g**, Positions of the follower relative to the leader. Lateral (**h-k**) and forward (**l-o**) swimming speed of the follower as a function of the leader's position.



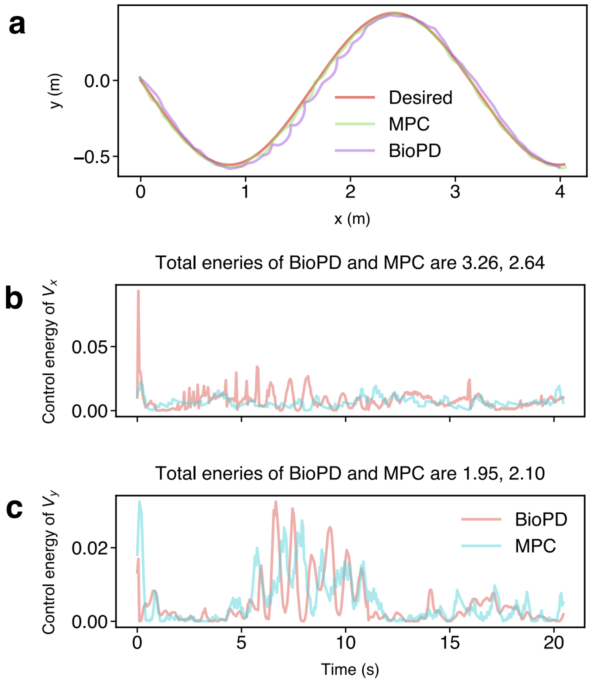
SUPPLEMENTARY FIGURE 14. Relative position differences between two virtual leaders and one following real fish in y- and z- axis. Distance difference between 2 virtual fish and real fish in the y-axis (**a**) in the local coordinate based on two virtual fish (origin is the average position of 2 virtual fish and points along the virtual fish head direction) and z-axis (**b**).



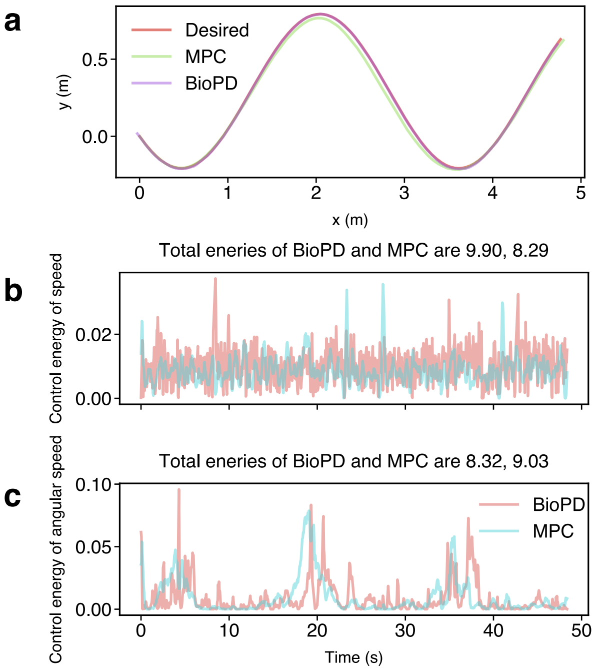
SUPPLEMENTARY FIGURE 15. Overview of experiments and data analyses with two virtual fish. a, Two virtual fish are swimming back and forth in an arena, with an average swimming speed of 0.04 m/s. They are positioned beside each other at a lateral distance of 0.08 m. **b**, The data were organized around a coordinate system with the origin at the centroid of the virtual fish's positions, and decisions were analyzed along the axis perpendicular to their direction of motion. **c**, An example of the real fish's position density in relation to the virtual fish. We calculated the normalized marginal probability distribution of the real fish's position (perpendicular to the virtual fish's movement direction) and stacked these distributions for various lateral distances between the virtual fish. **d**, We are able to maintain the direction of the virtual fish's movement without losing information because the real fish typically maintains a relatively stable front-to-back distance with its virtual conspecifics (**e**).



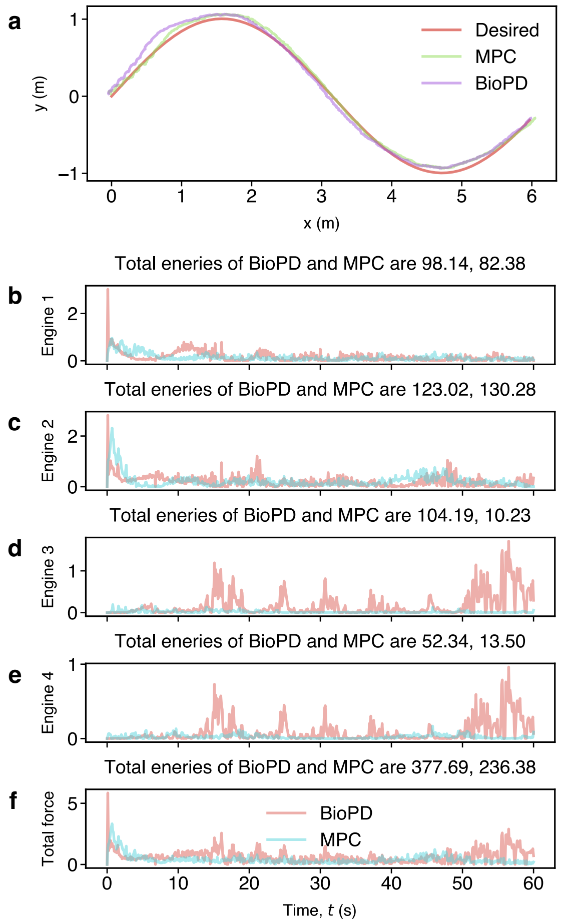
SUPPLEMENTARY FIGURE 16. Schematic to show how the critical distance l_c is determined based on the heatmap and piecewise function. a, Peaks are determined at each left-right distance, l. **b**, We flip the peak values along $x = 0$. **c**, We fit l_c according to the piecewise function.



SUPPLEMENTARY FIGURE 17. Comparison of the performance and control energy of the BioPD and MPC controllers for the Crazyflie drone. a, The trajectories of the drone controlled by each method. **b-c**, The control energy of the speed in the x (**b**)- and y (**c**)-axes, respectively.



SUPPLEMENTARY FIGURE 18. Comparison of the performance and control energy of the BioPD and MPC controllers for the PiCar Robot. a, The trajectories of the car controlled by each method. **b-c**, the control energy of the speed (**b**) and angular speed (**c**).



SUPPLEMENTARY FIGURE 19. Comparison of the performance and control energy of BioPD and MPC controllers for the robot boat. a, The trajectories of the boat controlled by each method. **b-e**, The control energy of the four engines within the boat. **f**, Total control energy of the four engines.

Trail description	Number of fish	Length of each trail
2 real fish	114 (57 pairs)	40 mins
2 real fish in the Matrix	48 (24 pairs)	60 mins
1 virtual fish at different speeds	39	90 mins
1 virtual fish bout vs constant speeds	22	90 mins
1 virtual fish at different display frequency	25	90 mins
1 virtual fish visibility control	32	90 mins
1 virtual fish in the Matrix to verify model	20 (10 pairs)	90 mins
2 virtual fish at different lateral distances and speeds	198	90 mins

Table 1: Number of zebrafish for each experiment.

1 References

1. Giernacki, W., Skwierczynski, M., Witwicki, W., Wronski, P. & Kozierski, P. Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering. *2017 22nd International Conference on Methods and Models in Automation and Robotics (Mmar)* 37–42 (2017).

2. Wang, W. *et al.* Design, modeling, and nonlinear model predictive tracking control of a novel autonomous surface vehicle. *2018 IEEE International Conference on Robotics and Automation (ICRA)* 6189–6196 (2018).